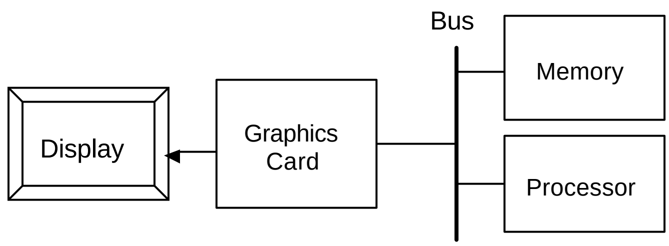
Networked Multimedia: Graphics



Lecture Overview

- Properties of human vision
- PC display architecture
- Graphics card technology
- Graphics software

Color in Human Vision

- Human eye has two types of sensors
  - rods: monochrome
    - high resolution
  - cones: respond to one of three colors (RGB)
    - lower resolution
    - other colors in light spectrum seen because they stimulate two colors at once (*e.g.* cyan = G & B)
    - so we can display all colors by adding RGB
  - color component can be less precise than the monochrome image - not noticed
    - color TV takes advantage of this; less bandwidth devoted to color
    - in general computer displays color every pixel

Resolution of Human Vision

- Visual persistence causes impression of constant or moving image if frame rate is above 25 Hz
  - but psychological fatigue is experienced up to 60 Hz
  - some displays run at 72 Hz and above to avoid this
- Center of vision field has much greater acuity
  - can be used to reduce data rate elsewhere in the scene
  - if you can be sure the audience will look where you want them to...

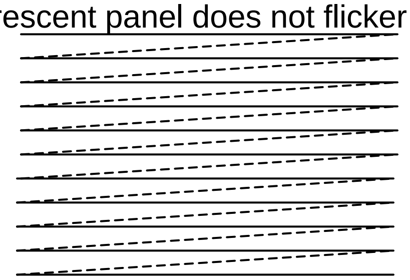
Perceptions in Vision

In general:

- Pixels blend into smooth image at the distance where they are too small to see individually
- Moving image perceived as better quality than still image of the same resolution
- Color images perceived as better quality than the same resolution monochrome
- Horizontal and vertical perceived differently
  - No stereo, and limited range, in vertical direction
  - Immersive effect greater if image occupies most or all peripheral vision
  - Thus CinemaScope movies and HDTV use wide wide aspect ratio (5:9 or greater versus regular 3:5)

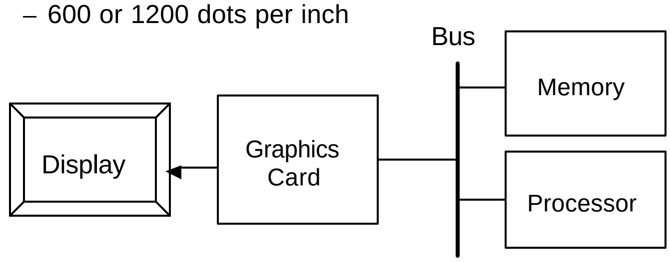
Raster Displays

- One result of visual persistence is that a CRT display can scan out the pixels progressively - called a *raster* - and the image is seen holistically
  - interlaced scan used by NTSC video also works but leads to more visual artifacts
- CRT must scan the three colors separately - colors add
- By contrast, LCD display is subtractive in nature
  - and the florescent panel does not flicker



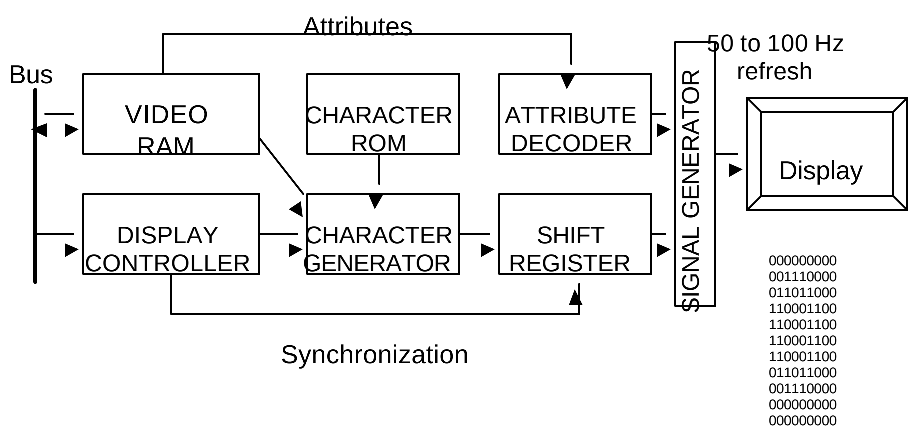
Graphics Card

- PC architecture offloads driving the display to a separate subsystem
  - allows modular, upgradeable configuration
    - 640x480, 800x600, 1024x768, 1280x1024, 1600x1200, 2048x1536...
    - more sophisticated designs and range of performance
  - combined with computer games market, has resulted in aggressive competition in the graphics card market
  - these principles apply to printers too, at higher resolution
    - 600 or 1200 dots per inch



Character Generation

- Legacy from original PC design
- Same basic technique used in software for graphic fonts



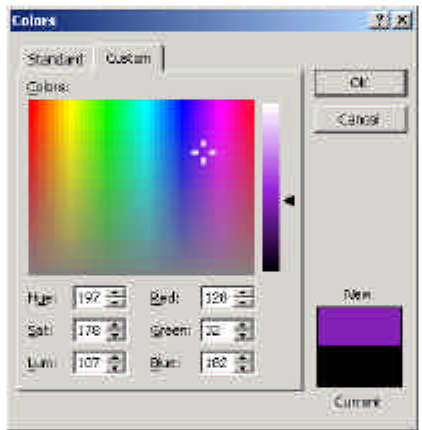
Graphics Mode

- Video RAM can be addressed as memory but also scans out in a raster pattern
- In character mode the video RAM holds character codes such as ASCII
- In video mode it holds the intensity and color of each pixel
- Pixels are in rectangular array and are "square"
  - same size in both dimensions
- The number of colors in the palette is a function of graphics card design
  - number of bytes per pixel influences color quality
  - more colors => more video RAM to hold codes for the same screen
  - typical: 8 bit (256 colors); 16 bit (64K colors); 32 bit (4096K colors)
  - standards from Video Electronics Standards Consortium (VESA)

Essential Computer Graphics Function

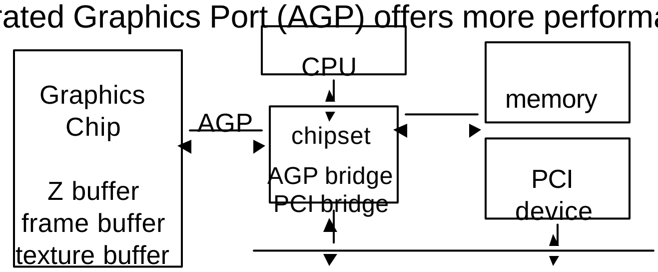
- All computer graphics consists of choosing the color and luminosity of every pixel in the rectangular screen
  - can be simple: pixels in a straight line
  - or very complex: moving 3D images
  - might "dither" pixels to achieve shading
    - a pixel alternates among two (or more) values
    - the average value is seen by the eye
- A huge amount of data is involved, so both acceleration and abstraction are applied
  - 1024 x 768 x 32 bits x 30 frames/second = 755 Mb/s
  - generally, major chunks of screen the same color
  - changing at relatively slow rate or can't be understood

Colors in Windows



AGP Graphics Cards

- Latest graphics cards have powerful accelerators onboard
  - really a special-purpose CPU
  - allows rendering of polygons and shapes with texture
  - support for 3D graphics
  - not useful unless the software is designed to use it
- Accelerated Graphics Port (AGP) offers more performance



Graphics Card Capabilities

- High performance processors on AGP cards allow sophisticated capabilities such as:
  - Alpha blending: transparency of objects
  - Background bitmap
  - Environment mapping: reflections in a reflective 3D object
  - Fogging: objects that disappear into a fog
  - Levels: multiple, switchable complete graphics ensembles
  - Polyhedrons: objects made of simple polygons
  - Radiosity: reflected light from illuminated surfaces
  - Rendering: filling and shaping surface of a 3D object
  - Specular highlights: dazzle on mettalic objects
  - Texture mapping: tiled micropatterns on surfaces

Graphics Software

- Programming graphics can be very very complex
- In the general case, we want 3D rendering of the virtual environment
  - generally accessed through a graphics environment
  - more on this later in the course
- Today most non-VE applications have a 2D GUI
  - Java has a *Frame* class for 2D graphic objects
    - there is a complex arrangement for fitting subordinate graphic objects into a space
    - text labeling, data entry blanks, colors etc can be configured
  - The Tool Command Language and Tool Kit (Tcl/Tk) developed by John Osterhout offers a higher-level graphics scripting environment
    - short learning curve and very fast development

Some Java GUI Code
from the NEW Floor Control

```
Object[ ] gridButtons;          // username buttons on the grid
 String[ ] gridNames;           // usernames associated with grid buttons
 TextArea receiveText;        // sendToAll and sendToController text receive area
 TextField enterUsername, enterPassword, enterCourseID, sendUrl;
 TextField systemMessages, sendToAll, sendToController; // one-line text areas
 Button requestFloor, changeFloorRules, grantNext;      // top row of buttons
 Button urlKill, urlNext, urlPrev;                // only when there is a URL file
 Choice urlDropdownList;                          // choose from URLs in file
 Color infoPanelColor;
 Panel topLevelPanel;        // all the other panels are added to this one
 Panel loginPanel;            // first panel to display; accepts
   username/password/courseID
 Panel floorControlPanel, floorRequestButtonPanel, namesGridPanel,
   messagesPanel;
 Panel urlDropdownPanel, urlButtonsPanel;
 Panel floorRequestPanel;     // grid of usernames, to be attached to
   floorControlPanel;
 CardLayout topLevelLayout;
```


Defining a Login Panel

```
// build login panel
   topLevelPanel = new Panel();
   topLevelLayout = new CardLayout();
   topLevelPanel.setLayout(topLevelLayout);
   enterUsername = new TextField(8);
   enterPassword = new TextField(8);
   enterCourseID = new TextField(11);
   enterPassword.setEchoChar('*');
   enterUsername.addActionListener(this);
   enterPassword.addActionListener(this);
   enterCourseID.addActionListener(this);
   loginPanel = new Panel();
   buildLoginPanel();
   topLevelPanel.add(loginPanel,"login");
   floorControlPanel = new Panel();
   topLevelPanel.add(floorControlPanel,"floor control");
   add(topLevelPanel);
```

Assembling the Login Panel

```
public void buildLoginPanel()
 { // basic panel structure
  loginPanel.setBackground(new Color(100,150,200));
  GridBagLayout loginPanelLayout = new GridBagLayout();
  loginPanel.setLayout(loginPanelLayout);
  GridBagConstraints loginPanelConstraints = new GridBagConstraints();
  loginPanelConstraints.fill = GridBagConstraints.NONE;
  // data fields
  Label usernameLabel = new Label("NAME",Label.LEFT);
  Label passwordLabel = new Label("PASSWORD",Label.LEFT);
  Label courseIDLabel = new Label("COURSE",Label.LEFT);
  Label topLabel = new Label("",Label.LEFT);
  Label bottomLabel = new Label("",Label.CENTER);
  Label onethirdDownLabel = new Label("",Label.CENTER);
  Label twothirdDownLabel = new Label("",Label.CENTER);
  usernameLabel.setFont(new Font("SansSerif", Font.BOLD, 16));
  usernameLabel.setForeground (new Color (255, 255, 255));
  enterUsername.setBackground (new Color (200, 200, 100));
  enterUsername.setFont(new Font("SansSerif", Font.PLAIN, 14));
  etc.
```


A Tcl/Tk Interface

NEW Whiteboard Tool Selection

```
proc BuildTools {w} {

    global GWbdFont charset

    button $w.text -text "T" \
          -command "ToolSet $w text" -borderwidth 3 \
          -font -Adobe-Times-*-R-Normal-*-*-180-*-*-*-*-*-*
    pack $w.text -side top -fill x

    button $w.line -image bitmap_line -text "Line" \
          -command "ToolSet $w line" -borderwidth 3
    pack $w.line -side top -fill x

    button $w.arrow -image bitmap_arrow -text "Arrow" \
          -command "ToolSet $w arrow" -borderwidth 3
    pack $w.arrow -side top -fill x

    etc.
```